Sustainable Energy Technologies and Assessments 67 (2024) 103818

Contents lists available at ScienceDirect

Sustainable Energy Technologies and Assessments

journal homepage: www.elsevier.com/locate/seta

A hybrid optimized data-driven intelligent model for predicting short-term demand of distribution network

Md. Ahasan Habib, M.J. Hossain *, Md. Morshed Alam, Md. Tariqul Islam

School of Electrical and Data Engineering, University of Technology Sydney, Ultimo, 2007, NSW, Australia

ARTICLE INFO

Keywords:
Demand forecasting
Artificial intelligence
Deep learning
Mean absolute percentage error

ABSTRACT

An advanced deep learning-based framework is presented in this study, utilizing sequential neural architecture to enhance precision in short-term load forecasting of low-voltage distribution networks. A three-stage paradigm for precise forecasting is presented, beginning with a generalizing data preprocessing approach, followed by multivariate feature construction and selection, and finally model hyperparameter modification. The proposed model employs feature engineering and clustering techniques, with the former being used to process historical load data, electricity prices, and ecological variables (temperature, dew point, wind speed, and humidity), and the latter, to extract highly correlated features as final inputs. The model's robustness is ensured by careful exploration and optimization of hyperparameters, and the model after post-optimization achieves a notable Mean Absolute Percentage Error (MAPE) of 0.57%, 0.99%, and 1.2% for 5, 15, and 30 min ahead forecasts, respectively. A detailed comparison with other deep learning algorithms reveals that the suggested model consistently outperforms them in anticipating load demands at different time intervals. This designed approach not only highlights the impact of the presented data-driven model but also conveys useful ideas to strengthen energy management in distribution networks.

Introduction

Accurate load forecasting benefits energy distributors and consumers by facilitating grid-side energy management and effective electrical network operation. There are several reasons why precise forecasting is necessary, for example efficient planning of power generation, managing energy, predicting prices, incorporating renewable energy sources, and many more [1,2]. The process of load forecasting can be broken down into three distinct categories, each of which has a distinct time horizon: ultra-short-term load forecasting (U-STLF), short-term load forecasting (STLF), and long-term load forecasting (LTLF) [3]. STLF is concerned with predicting the demand for a specific time frame, typically ranging from a few hours to several days. On the other hand, U-STLF involves predicting power consumption for a shorter time frame, usually ranging from a few minutes to several hours. In contrast, long-term forecasting predicts electricity demand spanning months to years. According to [4–7], U-STLF is frequently utilized in demand response programs, STLF is employed by utility businesses to ensure a steady and cost-effective supply of electricity, and LTLF is extensively used in power system planning and investment decisions. As renewable energy sources are deployed more often, precise ultra-short-term and short-term demand forecasting is becoming more and more important for maintaining grid stability and dependability. Therefore, U-STLF and STLF have garnered the attention of a significant number of studies that are now being conducted in the power industry.

To predict electricity demand, there are two main types of models: single-unit models that only make predictions, and hybrid models that combine an optimization unit with a dependable load forecasting model. Single-unit models use time-series forecasting to anticipate energy demand, but they typically underperform for nonlinear load patterns [8]. However, hybrid models enhance accuracy through feature engineering and optimization procedures, offering a complexity-precision trade-off for authorities to choose from. In the beginning, load forecasting relied on statistical methods, which were considerably upgraded by computer technology. This led to the development of models and algorithms such as ARIMA (AutoRegressive Integrated Moving Average), SARIMA (Seasonal ARIMA), ES (Exponential Smoothing), MLR (Multiple Linear Regression), SMA (Simple Moving Average), and WMA (Weighted Moving Average). These strategies improved the comprehension of the seasonal and temporal fluctuations in electricity demand by utilizing computing power. In [9], ARIMA and SARIMA frameworks are proposed for Israel's energy demands, with the optimal model possessing a MAPE exceeding 10%. Another study in Ref. [10] used three models to predict the load in South Korea and found that the

* Corresponding author.
E-mail address: jahangir.hossain@uts.edu.au (M.J. Hossain).

https://doi.org/10.1016/j.seta.2024.103818
Received 24 December 2023; Received in revised form 24 April 2024; Accepted 18 May 2024
Available online 31 May 2024

best MAPEs were between 2.35% and 6.39%. [11] highlights SARIMA's seasonal trend capture, while Ref. [12] discusses statistical approaches in load forecasting and their widespread application.

Machine Learning (ML) and Artificial Intelligence (AI)-based algorithms have played a significant roles in improving the accuracy of forecasting. New data-driven techniques are gaining popularity in academia for their ability to predict variable load patterns using self-training capabilities on available data, especially for seasonal or recurring energy usage patterns [13]. Some popular ML methods used for load forecasting are Random Forest (RF), Support Vector Regression (SVR), Reinforcement Learning, Extreme Gradient Boost (XGBoost), Gated Recurrent Unit (GRU), and others. An optimized model with a MAPE range from 1%–3% was obtained by using SVR for short-term energy consumption forecasting in Ref. [14–17]. According to [18], SVR is a good method for estimating load due to its resistance to outliers, quick decision model updates, and ease of generalization. Recently, Artificial Neural Network (ANN) techniques have been successful in predicting various loads, such as air compressor load [19], distribution substation load [20], university campus load [21], residential unit load [22], and others [23,24]. Although models for single-stage load forecasting are very accurate, hybrid models that combine optimization and feature engineering with forecasting provide increased robustness and dependability when dealing with unpredictable input data.

As mentioned earlier, a hybrid model with single-stage models, feature engineering, and optimization, ensures data relevance and optimal load prediction model's hyperparameter values. For example, Ref. [25–27] employed ML-based models and hyperparameter optimization to accurately predict load demand. The authors in [25] utilized Sparrow Search, [26] utilized Particle Swarm Optimisation (PSO), and [27] utilized the Firefly Optimization (FFO) Algorithm to fine-tune the hyperparameters of the SVM model to improve the accuracy of load forecasting. These optimizations improved accuracy but increased computational burden. Some researchers also used different special ML algorithms for forecasting the aggregated/grid load [28,29] or residential load [30]. Article [28] describes an approach using Modified Mutual Information (MMI) for data preparation and feature selection. After feature extraction and noise reduction, the Factored Conditional Restricted Boltzmann Machine (FCRBM) captures complicated non-linear correlations in load data for accurate load forecasts. At last, the Genetic Wind Driven Optimisation (GWDO) method optimizes model parameters for maximum performance. Projected outcomes show good performance, but framework complexity and absence of training and testing time information are major limitations. According to [29], the Wavelet Transform (WT) was utilized to consider time-series load data and apply Extreme Machine Learning (EML) for load estimation. After combining the series mathematically, the authors achieved an accurate model with a MAPE of 0.6%. The article compares different Neural Network (NN) models but does not show how to formulate the best model.

In [31], a hybrid model was utilized that includes Variational Mode Decomposition (VMD) and NN to predict load and prices in an isolated microgrid. VMD collects Intrinsic Mode Functions (IMFs) to reflect load characteristics in the load forecasting model, and the Gravitational Search Algorithm (GSA) optimizes model parameters to improve prediction accuracy. However, the weather and calendar data were not taken into account. Similar techniques were used in Ref. [32, 33], where real-time load data, temperature, and day-type information were combined with VMD for feature extraction. In [32], the hybrid model combines concurrent LSTM models with a Bayesian Optimization Algorithm (BOA) for model optimization. Several load forecasting algorithms, including CNN-LSTM [34,35], DRNN-LSTM [36], CNN-LSTM-BiLSTM [37], *etc.*, are designed based on hybrid deep learning algorithms, with every part serving a distinct purpose in making predictions. For example, the authors of Ref. [34] used LSTM to predict future load demand after extracting the load pattern using CNN. Additionally, the researchers utilized NN algorithms including ANN [38], DNN [39], ENN [40], WNN [38], *etc.* to monitor the trend of load consumption because NN is capable of remembering previous patterns like the human brain. To improve the accuracy of their forecasts, they implemented various clustering techniques [35,38,41]. Although the output of these hybrid models is sometimes superior, they can suffer from overfitting when fed new data. Different strategies are applied to mitigate common problems related to deep learning modes such as limited data availability in the literature. The authors in Ref. [42] addressed the issue of limited data in Greece's National Natural Gas Transmission System by providing a new correlation coefficient technique to reduce pseudo-correlation hazards and improve forecasting precision. Ref. [43] concentrated on transfer learning in energy systems, employing a hybrid method for selecting the transfer domain. The proposed WM algorithm combined Wasserstein distance and maximal information coefficient to create the WM-DSSFA-LSTM-TL model, which mitigated negative transfers and improved predictive performance in limited data/resource problems. Table 1 lists contemporary load forecasting approaches employing machine learning and deep learning hybrid models with notable contributions and research gaps/disadvantages.

The selection of input factors influences ML and DL models' capacity to discover patterns and generate correct predictions. The literature demonstrates that there is a lack of understanding and require further analysis in simultaneously addressing elements such as pricing, weather, and calendar impacts in load forecasting, even though these factors have a major impact on power consumption. Furthermore, as Refs. [44–46] explain, hyperparameter selection is crucial and demands careful testing for particular scenarios. The framework and input variables for some load forecasting models are listed in Table A1 (See Appendix A).

The objective of this study is to build a practical and complete power consumption model that can effectively handle the complicated nature of load forecasting. In order to be feasible, the proposed load prediction system uses real-time price data and meteorological data. Correlation analysis is used to identify highly correlated inputs, which reduces the overall training time. The study also uses optimizer tools and expert-driven analysis to make the model more efficient. The following is a summary of the research contribution:

- Estimate electricity demand using a realistic model that accounts for all relevant factors.
- Execute correlation analysis to select highly associated inputs to reduce training time and complexity.
- Consolidate optimizer tools with expert-driven exploration to augment the efficiency of the model.
- Test the model with real-time load data for short-term predicting accuracy.

The subsequent sections provide an in-depth analysis of the model's description and methodology, after which the results and analysis are thoroughly examined. The concluding discussion with possible applications is included in the final section of the paper.

Methodology & model description

This section systematically presents a hybrid Sequential Data-Driven Model (SDDM) for forecasting 5, 15, and 30-minute-ahead load demand. The model, shown in Fig. 1, combines advanced deep learning and clustering-based methods for more accurate load prediction. The procedure is broken down into four independent phases which are interconnected. These phases are as follows: (i) Data collection and aggregation; (ii) Data processing and analysis; (iii) Model development and optimization; and (iv) Performance evaluation. Each of these phases is discussed in the following sections.

Table 1
Recent excellent load forecasting models including achievements and research gaps.

Ref.	Model name	Opt. method	Repository	Key contributions/Achievements	Research gap
[28]	FCRBM	GWDO	USA power grid	• Modified feature selection technique is used. • Hybrid optimization method helps fast convergence. • Outperforms other deep learning models.	• Hybrid model and optimization technique reduce model realization. • Large execution time due to complex model.
[29]	EML	Not mentioned	ISO New England	• Alleviate the overtraining and uncertainty problems • Adaptability to seasonal variations	• Ensemble approach uses 24 parameters after WT, increasing model complexity. • Large execution time due to complex model.
[31]	ANN	GSA	PMJ & Favignana Island	• Good accuracy. • Generate one IMF from the preceding 200-hour data point.	• Rules for feature selection are unclear. • Electrical price is not considered here.
[32]	VMD-LSTM	BOA	Hubei Province, China	• Nonlinear mapping is used for feature selection. • Adaptability to seasonal variations	Instead of one model for all seasons, consider multiple.
[35]	CNN-LSTM	Not mentioned	Australian 69 Household data of SGSC project	• Cluster the households based on energy consumption. • Detailed case studies for different scenarios.	Hyperparameter optimization information is absent.
[37]	CNN-LSTM-BiLSTM	Not mentioned	A park in North China	• CNN and attention block extract relevant data features.	• The model is highly complicated. • Runtime is lengthy.

Data collection and aggregation

One of the most important tasks that must be completed during the first phase is the collection of a wide range of input data from several different databases. These databases each present themselves in a different format, such as CSV, Excel, or JSON. In this study, historical load data, calendar data, weather station data, and central electricity market operator price data are combined into an Initial data cloud. To ensure consistency, data from diverse sources in different forms is translated into CSV format and aggregated by date, time, and other parameters.

Data processing and analysis

After the data has been acquired, it is subjected to a series of careful processes, including cleaning, normalization, and transformation. Eq. (1) provides the mathematical expression for the Z-score approach, which is used to detect and correct anomalies. This method marks data points as outliers if their standard deviation from the mean is more than 3 standard deviations. One of the biggest challenges after identifying outliers is handling missing values and NaNs (Not a Number) entries. Eq. (2) enables linear interpolation to estimate missing values (x_{missing}) within a range (x_a and x_b) based on a target value. To ensure data continuity and completeness, Eq. (3) suggests filling NaNs ($x_n(\text{NaN})$) by calculating their average from the preceding $n-1$ data points.

$$Z = \frac{(X-\mu)}{\sigma}$$

$$\text{Outliers} = \{x \mid |Z(x)| > 3\} \tag{1}$$

$$x_{\text{missing}} = x_a + \frac{(x_b - x_a)}{(b-a)} \times (target - a) \tag{2}$$

$$x_n(\text{NaN}) = \text{avg}(x_{n-1}, x_{n-2}, x_{n-3}, \dots, x_{n-10}) \tag{3}$$

Selecting the most significant variables for predictive analysis is essential to building efficient ML models. When evaluating the linear relationship between two variables, Pearson's Correlation Coefficient (PCC) equation is shown in Eq. (4).

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i \text{ and } \bar{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$$
$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \tag{4}$$

This methodology employed clustering techniques on the preprocessed dataset after reducing its dimensionality using PCC. This approach established a more straightforward foundation for subsequent clustering by emphasizing the diversity in the dataset and identifying strong groups. This step improved the clustering algorithms' efficiency and effectiveness by allowing them to identify distinct groups with higher accuracy in a reduced-dimensional space, where the most informative data was now more visible and less affected by noise. The algorithm that is utilized for the clustering is presented in the following manner:

Model development and optimization

Model description and hyperparameter optimization describe how to utilize the dataset after clustering, which is divided into training, validation, and testing parts. This divide is critical for measuring Sequential Data-Driven Model (SDDM) robustness and generality. Through the integration of several single Data-Driven Units (DDUs) in a cascade arrangement, the proposed SDDM offers a unique approach to predicting future demand. Each DDU in this framework processes sequential data independently, and the outputs are used as inputs by the following layer, resulting in a tiered and interconnected structure. The complete SDDM can be represented by an array of n-numbered DDUs:

$$\text{SDDM} = [\text{DDU}^{(1)}, \text{DDU}^{(2)}, \text{DDU}^{(3)} \dots, \text{DDU}^{(n)}] \tag{5}$$

where n is the number of layers and the schematic diagram of DDU is depicted in Fig. 2. Appendix B contains a comprehensive explanation of the DDU's operational principle as well as its corresponding mathematical equations.

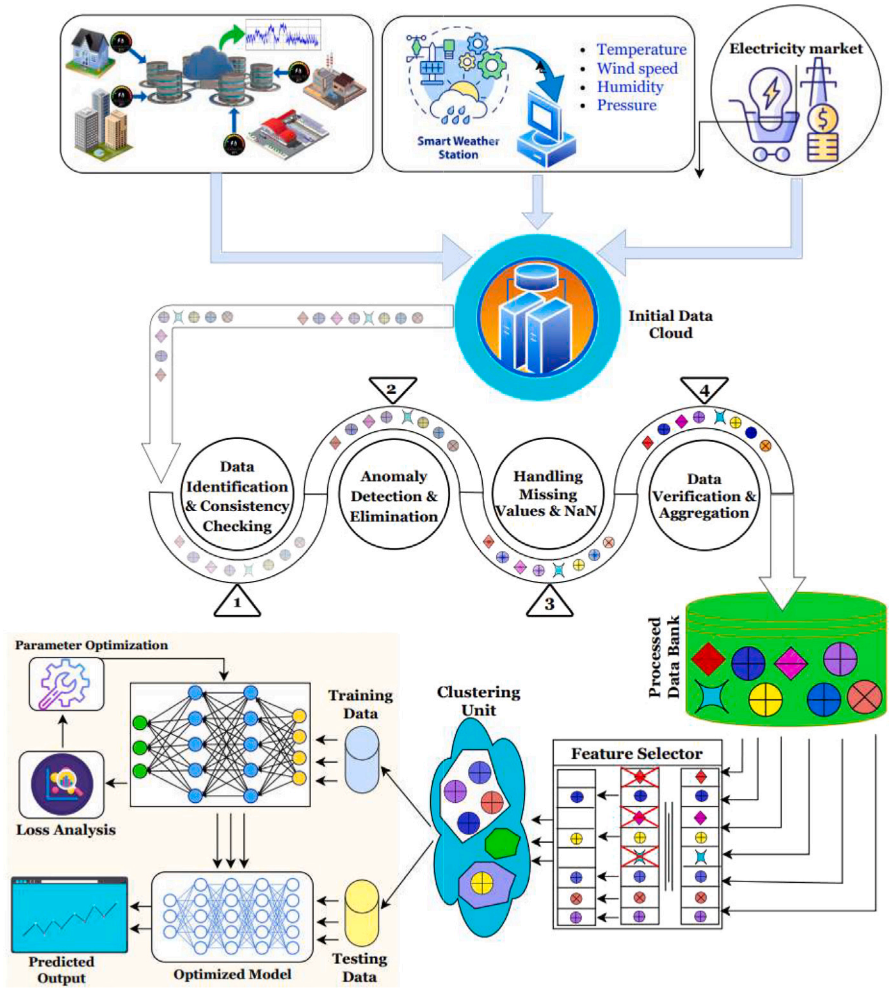

Fig. 1. Methodological framework for accurate load prediction by using the proposed SDDM.

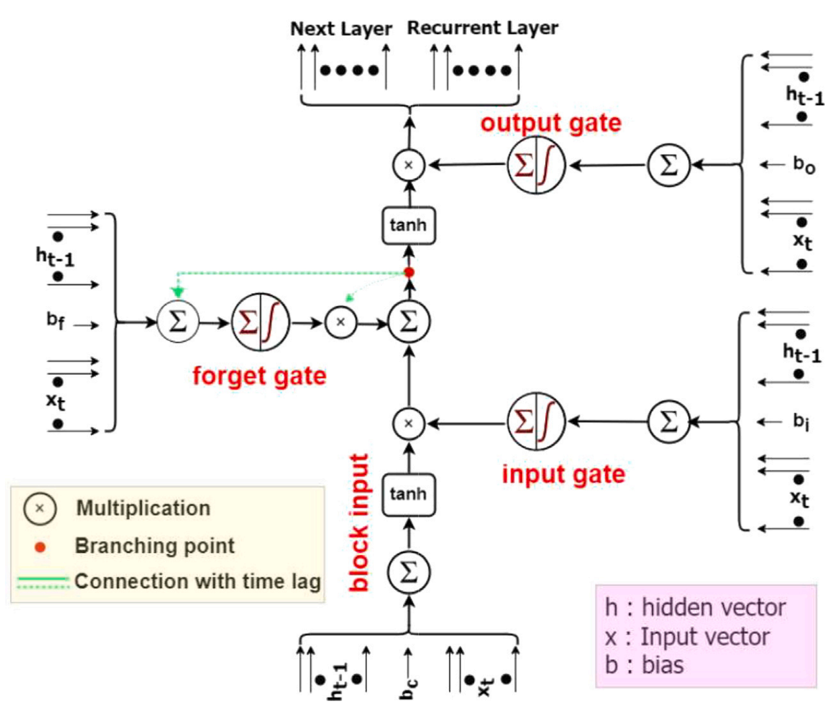

Fig. 2. Schematic diagram of single unit DDU of the proposed SDDM.

Algorithm 1 Machine learning-based clustering algorithm.

function CLUSTER(D, K, num_iterations)
  Select K unique data points randomly from D as the initial centroids $c_1, c_2, ..., c_K$
  Initialize cluster assignments $a(i)$ for each data point x_i in D
  for $i \leftarrow 1$ **to** num_iterations **do**
    // Assignment step
    for each point x_i in D **do**
      $a(i) \leftarrow \arg\min_j \|x_i - c_j\|^2$
    end for
    // Update step
    for each cluster $j = 1$ **to** K **do**
      $c_j \leftarrow \text{mean}(\{x_i \mid a(i) = j\})$
    end for
    // Check for convergence
    if centroids c_j have not changed significantly **then**
      break
    end if
  end for
  return clusters and centroids
end function

Optimizer for DDU

Adaptive Moment Estimation was chosen as the best algorithm for the DDU because it updates parameters quickly during training. Algorithm 2 provides an in-depth breakdown of the precise sequential process used by the optimizer to optimize the proposed DDU. Complete mathematical formulae for the optimizer are contained in Appendix C.

Algorithm 2 Steps of the Optimizer designed for SDDM.

function OPTIMIZER1($\theta, \alpha, \beta_1, \beta_2, \epsilon$, num_iterations)
  $m_0 \leftarrow 0$ ▷ Initialize biased first moment estimate
  $v_0 \leftarrow 0$ ▷ Initialize biased second moment estimate
  $t \leftarrow 0$ ▷ Initialize time step
  for $i \leftarrow 1$ **to** num_iterations **do**
    $t \leftarrow t + 1$
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla J(\theta_t)$
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla J(\theta_t))^2$
    $\hat{m}_t \leftarrow \frac{m_t}{1-\beta_1^t}$
    $\hat{v}_t \leftarrow \frac{v_t}{1-\beta_2^t}$
    $\theta_{t+1} \leftarrow \theta_t - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}$
  end for
  return θ_{final}
end function

Model hyperparameter optimization

To get the optimum model for forecasting future load demand, the hyperparameters of the proposed data-driven model are tuned during the training phase using the approach outlined in Algorithm 3. This method was specifically developed to systematically explore and modify hyperparameters using random search techniques, offering an effective strategy for exploring the hyperparameter space and enhancing the model's performance. Algorithm 3 details the optimizer's operations, highlighting the model's performance improvement through hyperparameter optimization.

Performance evaluation

The performance of the proposed model is assessed using several performance metrics, such as Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Squared Error (RMSE), and R-squared (R^2) score. The mathematical expression of the performance indicators are given below:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

The MAE calculates the average absolute difference between the predicted and actual values. MAPE is a relative error measurement that calculates the average percentage difference between estimated and actual values. The square root of the average of squared discrepancies between the expected and actual values is represented by RMSE. The R^2 value calculates the percentage of the dependent variable's variation that the independent variables can account for. The R^2 value represents the goodness-of-fit of the model and ranges from 0 to 1, with a number closer to 1 reflecting a better fit.

Algorithm 3 Hyperparameter optimization of intelligent forecasting model.

function HYPEROPT(hyperparameter_search_space, num_trials)
  tuner ← RandomSearchTuner(hyperparameter_search_space)
  tuner.search(num_trials)
  best_configuration ← tuner.get_best_hyperparameters()
  return best_configuration
end function
function RANDOMSEARCHTUNER(search_space)
  tuner ← RandomSearch(model_builder, objective = *val_loss*) ▷ Minimizing validation loss
  tuner.search(search_space, epochs = num_epochs, validation_data = val_data)
  return tuner
end function
function RANDOMSEARCH(search_space, num_iterations)
  best_configuration ← None
  best_metric ← +∞
  for $i \leftarrow 1$ **to** num_iterations **do**
    current_configuration ← sample_uniformly(search_space)
    current_metric ← evaluate_performance (current_configuration)
    if current_metric < best_metric **then**
      best_configuration ← current_configuration
      best_metric ← current_metric
    end if
  end for
  return best_configuration
end function
function SAMPLE_UNIFORMLY(search_space)
  return randomly_sample_from(search_space)
end function
function EVALUATE_PERFORMANCE(configuration)
  return model_performance(configuration)
end function

Result and discussion

Data collection

The electricity load demand and pricing statistics were acquired from the Australian Energy Market Operator (AEMO), with a specific focus on the New South Wales (NSW) region in Australia. The dataset includes data from September 1, 2022, to February 28, 2023, showing different load trends and market changes. Meteorological data from the Australian Bureau of Meteorology (BoM) for the same period was also used. After gathering all data from the listed sources, they are processed using the approach outlined in Section "Data processing and analysis". This included temperature, humidity, wind speed, and other factors that affect power consumption. The load forecasting model was made with Python and Keras and TensorFlow as backend tools. TensorFlow is a well-known deep learning library, and Keras is a high-level neural network API that sped up the model-building process.

Input features selections

Following the data preparation, the subsequent task is to identify the features that exhibit a strong connection with output among a variety of choices to put in the SDDM. The input attributes are considered to be closely correlated with the demand, which is the target variable. Figure A1 displays the correlation coefficient between the target and input variables. The input (x_n) and output matrix of the proposed load forecasting model are shown below where, D= demand, M= minute, W= no of week in the month, WE= weekday (0) or weekend (1), MY= month of the year, DM= day of the month, C= cluster, and P= electricity price.

$$x_n = \begin{bmatrix} D_{(t=0)} & M_{(t=0)} & W_{(t=0)} & WE_{(t=0)} & MY_{(t=0)} & DM_{(t=0)} & C_{(t=0)} & P_{(t=0)} \\ D_{(t=1)} & M_{(t=1)} & W_{(t=1)} & WE_{(t=1)} & MY_{(t=1)} & DM_{(t=1)} & C_{(t=1)} & P_{(t=1)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ D_{(t=n)} & M_{(t=n)} & W_{(t=n)} & WE_{(t=n)} & MY_{(t=n)} & DM_{(t=n)} & C_{(t=n)} & P_{(t=n)} \end{bmatrix}$$

Output, $y_{n+1} = \mathrm{SDDM}(x_n)$

The recommended load forecasting model employs a cascade connection of n number of DDUs, and the same dataset is used for three separate case studies (5 min, 15 min, and 30 min ahead of time). In each case study, several hyperparameters varied to see how they affected the model's performance and pick the best sets from the alternatives. The following Table A2 presents the ranges for each of the following hyperparameters in all case studies: the number of DDUs, the number of neurons in each layer, the learning rate, the dropout rate, the batch size, the number of epochs, and the activation function.

Case study 1

This part extensively analyzes the model's performance and modifies each hyperparameter independently for 5 min ahead of demand forecasting which is represented in Figure A2. The optimizer first determines the initial model parameters, and in the subsequent discussion, the final model parameters are provided using a heuristic method.

(a) Number of neurons per layer selection: The performance of the proposed model with two hidden layers for determining the number of neurons per layer is illustrated in Figure A2a. The model was trained with a learning rate of 0.001, a batch size of 32, 75 epochs, no dropout, and a range of 64 to 192 neurons, increasing by 32 at a time. As illustrated in Figure A2a, the outcomes indicate that the performance metrics for each case are very similar but the performance is optimal with 128 neurons. Increased neuron numbers can help the model capture complicated connections and improve prediction, but they also increase overfitting risk. Thus, the proposed framework recommends 128 neurons per layer.

(b) Learning rate selection: Figure A2b indicates the impact of different learning rates on the performance of the proposed model. According to the graph, the learning rate of 0.001 resulted in the most accurate and reliable predictions. The same results were seen at a higher learning rate of 0.0012, however a higher learning rate speeds convergence but increases overfitting. Thus, the optimal learning rate is 0.001.

(c) Dropout rate selection: During training, the dropout rate randomly sets a percentage of neurons to zero to increase unpredictability and model robustness. However high dropout rates may hinder the model's learning and cause underfitting. Figure A2c shows that a dropout rate of zero results in the lowest prediction error, making it the optimal value.

(d) Batch size selection: The number of samples processed in each training cycle is determined by the batch size. Larger batch sizes can estimate gradients more accurately but require more memory, while smaller batch sizes may reduce convergence time. According to this analysis, a batch number up to 32 had lower MAE, MAPE, and RMSE values and higher R^2 scores (Figure A2d), but after that, accuracy started to go down so 32 is the best batch size for this model.

(e) Number of epochs selection The number of epochs determines how many times the model iterates over the training dataset. Although adding more epochs can help the model learn better, it should be done carefully to avoid overfitting. As depicted in Figure A2e, performance metrics exhibit an upward trend for a period of 75 epochs, after which they begin to decrease. So, it turns out that 75 epochs is the best number to use to train the model.

Based on the discussion above, the final value of hyperparameters for the proposed model is provided in Table A3 along with the value of performance metrics. The output of the proposed SDDM model for forecasting load demand is shown in Fig. 3 (only two weekdays and two weekends are provided for clarity). In the diagram, solid lines reflect load demand and dotted lines represent model output. The model is highly accurate and outperforms models discussed in literature review.

Case study 2

The model hyperparameters will now be separately optimized and displayed with the performance metrics for the 15-minute ahead load forecasting model. The optimal model hyperparameters and performance will be tabulated in the following discussion.

(a) Number of neurons per layer selection: Figures A3a and A3b illustrate the impact of varying the number of neurons on the proposed model utilizing the following parameters: a learning rate of 0.0008, a dropout rate of 0.2, a batch size of 24, and 100 epochs. The first diagram illustrates that the optimal number of neurons per hidden layer to attain the lowest MAPE, RMSE, and MAE values of 0.99%, 106 MW, and 77 MW, respectively, is 128. After changing the number of neurons in one layer while keeping the same number in the other, the prediction model will perform best with 128 neurons per layer, as seen in Figure A3b.

(b) Learning rate and dropout rate selection: The suggested model's performance metrics are shown in Figure A3c and A3d, respectively, for varied learning rates and dropout rates. The findings show that raising the learning rate enhances performance up to a point of 0.0008, beyond which additional increases cause performance to deteriorate. Similar to this, performance metrics increase with dropout rates up to 0.3 before they begin to decline. The dropout rate of 0.3, which provides equivalent outcomes to a dropout rate of 0.2 but with shorter training time, is selected as the best alternative after taking into account the trade-off between performance and training time.

(c) Batch size and number of epochs selection: Figure A3e and A3f illustrate the findings from the investigation regarding the most effective batch size and number of epochs for the forecasting model. The results demonstrate that the performance indicators are optimum when the batch size is 24, which is considered to be the ideal batch size for the model. As the total number of epochs increases, performance metrics improve and reach their pinnacle at 100 epochs. After this, the values of RMSE, MAE, and R^2 mostly remain constant, while the MAPE score somewhat rises, making 100 epochs the best option for demand forecasting that is 15 min in advance.

Table A4 shows the final selected hyperparameter values that have been thoroughly investigated and optimized to maximize the forecasting model's performance. Fig. 4 compares projected and actual load demand to demonstrate the effectiveness of the optimized model. The graph illustrates the precision of the model in forecasting load demands by employing the chosen hyperparameters.

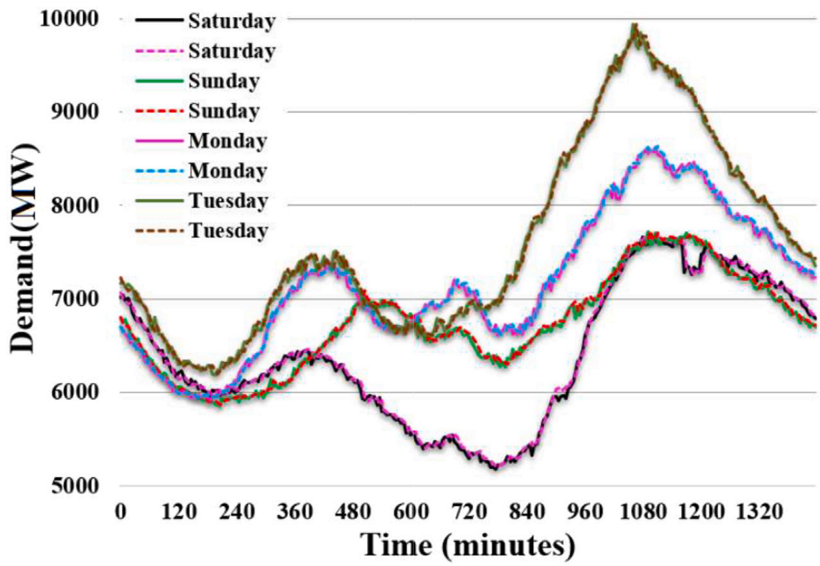

Fig. 3. Actual and five minutes ahead predicted load demand from the proposed U-STLF model.

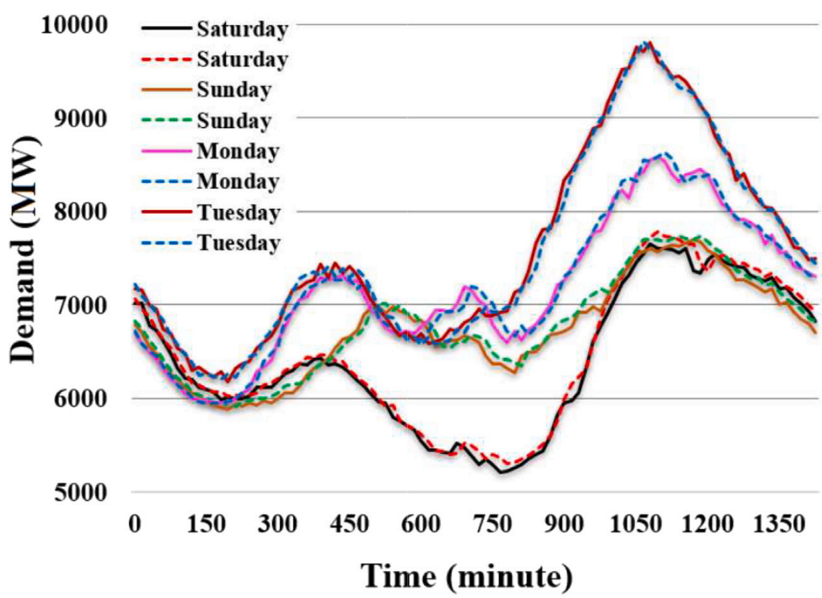

Fig. 4. Actual and predicted load demand from the proposed 15-minutes ahead STLF model.

Case study 3

In this section, the model performance for 30-minute forecasting will be investigated by separately adjusting the hyperparameters, and the optimized values will be tabulated.

(a) Number of neurons per layer selection: The number of neurons in the hidden layers was systematically increased from 64 to 192 with a step size of 32, and Figure A4a depicts the performance metrics over the range of 96 to 160 neurons to provide a clear overview of the findings. Up to 128 neurons, performance metrics are seen to be steadily improving, but after that point, they began to drop, and that is why it is taken as optimum. Individual analyses of the neurons in each layer (Figure A4b) also consistently confirmed these findings, emphasizing the idea that 128 neurons produce the best results.

(b) Learning rate and dropout rate selection: During the parameter tuning procedure, the learning rate is consistently altered between 0.0006, 0.0008, and 0.001, while the dropout rate is examined at values of 0.1, 0.2, and 0.3. The numerical values of the performance metrics are shown in Figures A4c and A4d which reveal that the learning rate and dropout rate's ideal values are found to be 0.0008 and 0.2, respectively.

(c) Batch size and number of epochs selection: Figures A4e and A4f illustrate the performance metrics that were collected through testing and analysis to choose the best hyperparameters for our model. Various batch sizes, including 40, 48, 56, and 64, as well as different numbers of epochs ranging from 50 to 150 with intervals of 25 epochs, were extensively studied. Based on the results, it was found that batch sizes of 56 and 100 epochs produced the best outcomes, outperforming all other hyperparameter combinations in terms of overall model performance.

The final hyperparameter values, which have been carefully examined and optimized to maximize the forecasting model's effectiveness, are shown in Table A5. Fig. 5 compares projected and real load demand to demonstrate the optimized model's effectiveness with the hyperparameters indicated in the table.

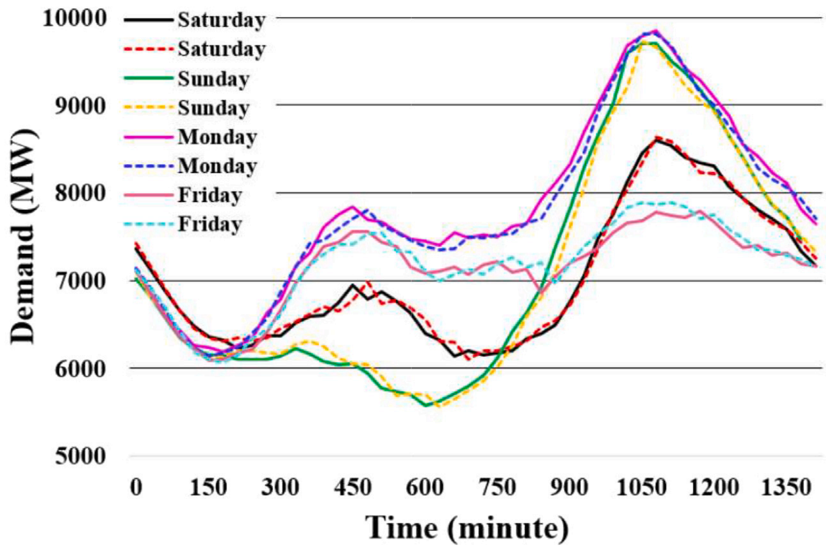

Fig. 5. Actual and predicted load demand from the proposed 30-minutes ahead STLF model.

Table 2
Performance Metrics Comparison of Different Models.

Model	MAPE [%]	RMSE [MW]	R^2 score	MAE [MW]
Vanilla LSTM	0.88	87.32	0.88	68.69
Bi-LSTM	2.49	218.07	0.96	183.64
CNN	4.28	410.44	0.87	343.73
CNN-LSTM	1.51	153.66	0.98	118.75
SDDM	**0.57**	**56.44**	**0.998**	**43.89**

Case study 4

This case study compares the accuracy of five-minute forecasts using a variety of machine learning models, including CNN, BiLSTM, CNN-LSTM, vanilla LSTM, and the proposed SDDM. This comparison is shown visually in Fig. 6, which makes it clear that the suggested model makes better predictions than other methods. The performance of the conventional LSTM, which is renowned for its ability to capture long-term dependencies, is comparatively inferior to that of the suggested model. However, it outperforms all other models except the proposed one. The CNN model, which is widely used to identify spatial patterns, performs the worst since the demand data is highly irregular and nonlinear, and it does not achieve the same level of accuracy as other benchmark methods. The Bi-LSTM model, which incorporates bidirectional information flow, also falls short of the suggested model in terms of accuracy. Finally, the CNN-LSTM model, which combines CNN and LSTM attributes, can accurately forecast future load but falls short of the SDDM. The comparative models' performance indicators are shown in Table 2 to validate these results. According to these metrics, the suggested model does better than the others in every aspect.

Apart from comparing the performance of forecasting five minutes ahead, the model performance is also thoroughly examined for slightly longer time horizons—specifically, fifteen and thirty minutes ahead. This comprehensive evaluation is carried out using the aforementioned deep learning models, and the results are shown in Figure A5 and Figure A6, respectively. Both figures indicate that the proposed SDDM's predicted outcomes are far better than the state-of-the-art models. The result again points out that the proposed model is better at capturing and predicting complex data patterns, proving its superiority as a forecasting tool.

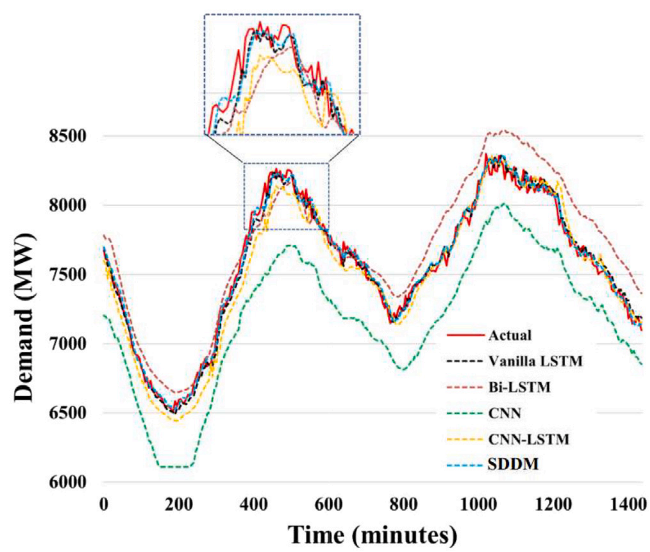

Fig. 6. Comparison of predicted demand by different machine learning models with the proposed one.

Conclusion

In this paper a deep learning-based intelligent model is developed that provides an effective way to forecast load by using real-time demand, pricing, weather, and calendar data as inputs. Multiple case studies at different time frames (5 min, 15 min, and 30 min) show that it consistently performs better than other popular deep learning-based models. The model offers the lowest errors with a greater temporal resolution, such as five-minute intervals, as it is capable of discerning more intricate patterns within the data. The optimizer tool thoroughly investigates the model's hyperparameters and validates them again to ensure its reliability. Its integration can help in a variety of ways, including the integration of Distributed Energy Resources (DER) and enhanced Energy Management Systems (EMS) to minimize Photovoltaic (PV) curtailment. The model can also aid in the detection of peak demand periods and make load balancing easier which can promote effective resource scheduling and grid management. Additionally, its use may make it easier to estimate future prices in dynamic tariff power system networks, facilitating efficient energy use and invoicing.

Funding information

This work was conducted without funding from external sources.

CRediT authorship contribution statement

Md. Ahasan Habib: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **M.J. Hossain:** Writing – review & editing, Supervision, Software, Resources, Project administration, Investigation. **Md. Morshed Alam:** Writing – review & editing, Visualization. **Md. Tariqul Islam:** Writing – review & editing, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.seta.2024.103818.

References

[1] Amasyali K, El-Gohary NM. A review of data-driven building energy consumption prediction studies. Renew Sustain Energy Rev 2018;81:1192–205.

[2] Gu W, Wu Z, Bo R, Liu W, Zhou G, Chen W, et al. Modeling, planning and optimal energy management of combined cooling, heating and power microgrid: A review. Int J Electr Power Energy Syst 2014;54:26–37.

[3] Singh AK, Ibraheem, Khatoon S, Muazzam M, Chaturvedi DK. Load forecasting techniques and methodologies: A review. In: 2012 2nd international conference on power, control and embedded systems. 2012, p. 1–10. http://dx.doi.org/10.1109/ICPCES.2012.6508132.

[4] Chen Y, Xu P, Chu Y, Li W, Wu Y, Ni L, et al. Short-term electrical load forecasting using the support vector regression (SVR) model to calculate the demand response baseline for office buildings. Appl Energy 2017;195:659–70.

[5] Dagdougui H, Bagheri F, Le H, Dessaint L. Neural network model for short-term and very-short-term load forecasting in district buildings. Energy Build 2019;203:109408.

[6] Zia MF, Elbouchikhi E, Benbouzid M. Microgrids energy management systems: A critical review on methods, solutions, and prospects. Appl Energy 2018;222:1033–55.

[7] Khuntia SR, Rueda JL, van Der Meijden MA. Forecasting the load of electrical power systems in mid-and long-term horizons: a review. IET Gener Transm Distrib 2016;10(16):3971–7.

[8] Nassif AB, Soudan B, Azzeh M, Attilli I, AlMulla O. Artificial intelligence and statistical techniques in short-term load forecasting: a review. 2021, arXiv preprint arXiv:2201.00437.

[9] Alberg D, Last M. Short-term load forecasting in smart meters with sliding window-based ARIMA algorithms. Vietnam J Comput Sci 2018;5:241–9.

[10] Ahn B-H, Choi H-R, Lee H-C. Regional long-term/mid-term load forecasting using SARIMA in South Korea. J Korea Acad-Ind Coop Soc 2015;16(12):8576–84.

[11] Zheng C, Wu Y, Chen Z, Wang K, Zhang L. A load forecasting method of power grid host based on SARIMA-gru model. In: Theoretical computer science: 39th National conference of theoretical computer science, NCTCS 2021, Yinchuan, China, July 23–25, 2021, revised selected papers 39. Springer; 2021, p. 135–53.

[12] Hammad MA, Jereb B, Rosi B, Dragan D, et al. Methods and models for electric load forecasting: a comprehensive review. Logist Sustain Transp 2020;11(1):51–76.

[13] Tavassoli-Hojati Z, Ghaderi S, Iranmanesh H, Hilber P, Shayesteh E. A self-partitioning local neuro fuzzy model for short-term load forecasting in smart grids. Energy 2020;199:117514. http://dx.doi.org/10.1016/j.energy.2020.117514.

[14] Hu Z, Bao Y, Chiong R, Xiong T. Mid-term interval load forecasting using multi-output support vector regression with a memetic algorithm for feature selection. Energy 2015;84:419–31.

[15] Vrablecová P, Ezzeddine AB, Rozinajová V, Šárik S, Sangaiah AK. Smart grid load forecasting using online support vector regression. Comput Electr Eng 2018;65:102–17.

[16] Chen Y, Xu P, Chu Y, Li W, Wu Y, Ni L, et al. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. Appl Energy 2017;195:659–70.

[17] Yang Y, Che J, Deng C, Li L. Sequential grid approach based support vector regression for short-term electric load forecasting. Appl Energy 2019;238:1010–21.

[18] Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A. A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing 2020;408:189–215.

[19] Wu D-C, Asl BB, Razban A, Chen J. Air compressor load forecasting using artificial neural network. Expert Syst Appl 2021;168:114209.

[20] Veeramsetty V, Deshmukh R. Electric power load forecasting on a 33/11 kv substation using artificial neural networks. SN Appl Sci 2020;2(5):855.

[21] Baek S-J, Yoon S-G. Short-term load forecasting for campus building with small-scale loads by types using artificial neural network. In: 2019 IEEE power & energy society innovative smart grid technologies conference. IEEE; 2019, p. 1–5.

[22] Dagdougui H, Bagheri F, Le H, Dessaint L. Neural network model for short-term and very-short-term load forecasting in district buildings. Energy Build 2019;203:109408.

[23] Román-Portabales A, López-Nores M, Pazos-Arias JJ. Systematic review of electricity demand forecast using ANN-based machine learning algorithms. Sensors 2021;21(13):4544.

[24] Nti IK, Teimeh M, Nyarko-Boateng O, Adekoya AF. Electricity load forecasting: a systematic review. J Electr Syst Inf Technol 2020;7(1):1–19.

[25] Guo Z, Hu L, Wang J, Hou M. Short-term load forecasting based on SSA-LSSVM model. In: 2021 4th international conference on energy, electrical and power engineering. 2021, p. 1215–9. http://dx.doi.org/10.1109/CEEPE51765.2021.9475790.

[26] Dai Y, Zhao P. A hybrid load forecasting model based on support vector machine with intelligent methods for feature selection and parameter optimization. Appl Energy 2020;279:115332.

[27] Hafeez G, Khan I, Jan S, Shah IA, Khan FA, Derhab A. A novel hybrid load forecasting framework with intelligent feature engineering and optimization algorithm in smart grid. Appl Energy 2021;299:117178.

[28] Hafeez G, Alimgeer KS, Khan I. Electric load forecasting based on deep learning and optimized by heuristic algorithm in smart grid. Appl Energy 2020;269:114915.

[29] Li S, Goel L, Wang P. An ensemble approach for short-term load forecasting by extreme learning machine. Appl Energy 2016;170:22–9.

[30] Bessani M, Massignan JA, Santos TM, London Jr JB, Maciel CD. Multiple households very short-term load forecasting using bayesian networks. Electr Power Syst Res 2020;189:106733.

[31] Heydari A, Nezhad MM, Pirshayan E, Garcia DA, Keynia F, De Santoli L. Short-term electricity price and load forecasting in isolated power grids based on composite neural network and gravitational search optimization algorithm. Appl Energy 2020;277:115503.

[32] He F, Zhou J, Feng Z-k, Liu G, Yang Y. A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm. Appl Energy 2019;237:103–16.

[33] Lv L, Wu Z, Zhang J, Zhang L, Tan Z, Tian Z. A VMD and LSTM based hybrid model of load forecasting for power grid security. IEEE Trans Ind Inf 2021;18(9):6474–82.

[34] Rafi SH, Deeba SR, Hossain E, et al. A short-term load forecasting method using integrated CNN and LSTM network. IEEE Access 2021;9:32436–48.

[35] Alhussein M, Aurangzeb K, Haider SI. Hybrid CNN-LSTM model for short-term individual household load forecasting. IEEE Access 2020;8:180544–57.

[36] Wen L, Zhou K, Yang S, Lu X. Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting. Energy 2019;171:1053–65.

[37] Wu K, Wu J, Feng L, Yang B, Liang R, Yang S, et al. An attention-based CNN-LSTM-BiLSTM model for short-term electric load forecasting in integrated energy system. Int Trans Electr Energy Syst 2021;31(1):e12637.

[38] Aly HH. A proposed intelligent short-term load forecasting hybrid models of ANN, WNN and KF based on clustering techniques for smart grid. Electr Power Syst Res 2020;182:106191.

[39] Hong Y, Zhou Y, Li Q, Xu W, Zheng X. A deep learning method for short-term residential load forecasting in smart grid. IEEE Access 2020;8:55785–97.

[40] Xie K, Yi H, Hu G, Li L, Fan Z. Short-term power load forecasting based on elman neural network with particle swarm optimization. Neurocomputing 2020;416:136–42.

[41] Sideratos G, Ikonomopoulos A, Hatziargyriou ND. A novel fuzzy-based ensemble model for load forecasting using hybrid deep neural networks. Electr Power Syst Res 2020;178:106025.

[42] Wei N, Yin L, Yin C, Liu J, Wang S, Qiao W, et al. Pseudo-correlation problem and its solution for the transfer forecasting of short-term natural gas loads. Gas Sci Eng 2023;119:205133.

[43] Wei N, Yin C, Yin L, Tan J, Liu J, Wang S, et al. Short-term load forecasting based on WM algorithm and transfer learning model. Appl Energy 2024;353:122087.

[44] Keskar NS, Mudigere D, Nocedal J, Smelyanskiy M, Tang PTP. On large-batch training for deep learning: Generalization gap and sharp minima. 2017, arXiv:1609.04836.

[45] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT Press; 2016, http://www.deeplearningbook.org.
[46] Bengio Y. Practical recommendations for gradient-based training of deep architectures. In: Neural networks: Tricks of the trade. 2nd ed.. Springer; 2012, p. 437–78.

Further reading

[1] Franklin JA. Jazz melody generation from recurrent network learning of several human melodies.. In: FLAIRS conference. 2005, p. 57–62.